

On absolute linear Harbourne constants

M. Dumnicki, D. Harrer, J. Szpond*

January 15, 2016

Abstract

In the present note we study absolute linear Harbourne constants. These are invariants which were introduced in [2] in order to relate the lower bounds on the selfintersection of negative curves on birationally equivalent surfaces to the complexity of the birational map between them. We provide various lower and upper bounds on Harbourne constants and give their values for the number of lines s of the form $p^{2r} + p^r + 1$ for any prime number p and also for all values of s up to 31. This extends considerably results of the third author obtained earlier in [10].

Keywords arrangements of lines, combinatorial arrangements, Harbourne constants, finite projective plane, bounded negativity conjecture

Mathematics Subject Classification (2000) 14C20; 52C30; 05B30

1 Introduction

Arrangements of lines were introduced to algebraic geometry by Hirzebruch in his papers concerning the geography of surfaces (i.e. construction of surfaces X with prefixed invariants $c_1^2(X)$ and $c_2(X)$), see [7], [1].

Multiplier ideals defined by arrangements of lines were studied by Teitler [11] and Mustaŭ [8].

Recently arrangements of lines appeared in the ideas revolving around the Bounded Negativity Conjecture (BNC for short), see [3] for the background of the Conjecture and [2], [9] for the role of configurations of lines. Whereas BNC is relevant only over a field of characteristic zero, some related problems are of interest over arbitrary fields. In [2] the authors introduced and began to study linear Harbourne constants. These are certain invariants computed by configurations of lines in the projective plane. Even though the Bounded Negativity fails in positive characteristic, it is clear from Definition 1.1 that for a fixed d , the linear Harbourne constant $H(d)$ is a finite number (because the number of combinatorial possibilities for invariants of a configuration of d lines is finite). It is interesting to estimate these numbers because in particular they measure the discrepancy between combinatorial data sets, see [4] and those sets which come from geometric configurations defined over some fields.

For the purpose of this note, a configuration \mathcal{L} is a finite set of mutually distinct lines $\mathcal{L} = \{L_1, \dots, L_d\}$. Given a configuration \mathcal{L} , we define its singular set $\mathcal{P}(\mathcal{L}) =$

*The first and the third authors were partially supported by National Science Centre, Poland, grant 2014/15/B/ST1/02197

$\{P_1, \dots, P_s\}$ as a set of points where two or more lines intersect. This is the same as the singular locus of the divisor $L_1 + \dots + L_d$. For a point $P \in \mathcal{P}(\mathcal{L})$, we denote by $m_{\mathcal{L}}(P)$ its *multiplicity*, i.e. the number of lines which pass through P . We have the following definitions.

Definition 1.1. The *linear Harbourne constant of a configuration of lines \mathcal{L} in the projective plane $\mathbb{P}^2(\mathbb{K})$* is the rational number

$$H(\mathbb{K}, \mathcal{L}) = \frac{d^2 - \sum_{k=1}^s m_{\mathcal{L}}(P_k)^2}{s}. \quad (1)$$

The *linear Harbourne constant of d lines over \mathbb{K}* is defined as the minimum

$$H(\mathbb{K}, d) := \min H(\mathbb{K}, \mathcal{L})$$

taken over all configurations \mathcal{L} of d lines.

Finally the *absolute linear Harbourne constant of d lines* is the minimum

$$H(d) := \min_{\mathbb{K}} H(\mathbb{K}, d)$$

taken over all fields \mathbb{K} .

In order to alleviate the notation we define first the set

$$Q = \{q = p^r, \quad p \text{ is prime}, \quad r \in \mathbb{Z}_{>0}\}.$$

For an integer d , we define $q(d)$ as the least number $q \in Q$ satisfying

$$d \leq q^2 + q + 1$$

and $r(d)$ as the largest number $r \in Q$ satisfying

$$r^2 + r + 1 \leq d.$$

Systematic investigation of absolute linear Harbourne constants $H(d)$ was initiated in [10]. Results stated there and computer supported experiments have led us to formulate the following conjecture.

Conjecture 1.2. For $d \geq 2$ let $q = q(d)$ and let $i := q^2 + q + 1 - d$.

If $i \leq 2q - 2$, then

$$H(d) = h(d)$$

where

$$h(d) = \frac{q^2 + q + 1 - i - \varepsilon_1(i)m_1(i) - \varepsilon_2(i)m_2(i) - t_{q-1}(i)(q-1) - t_q(i)q - t_{q+1}(i)(q+1)}{\varepsilon_1(i) + \varepsilon_2(i) + t_{q-1}(i) + t_q(i) + t_{q+1}(i)},$$

with

$$\begin{aligned} m_1(i) &= q + 1 - i, & m_2(i) &= 2q + 1 - i \\ \varepsilon_1(i) &= \begin{cases} 1 & \text{for } 0 \leq i \leq q-1 \\ 0 & \text{otherwise} \end{cases}, & \varepsilon_2(i) &= \begin{cases} 1 & \text{for } i > q+1 \\ 0 & \text{otherwise} \end{cases}, \\ t_{q-1}(i) &= \begin{cases} qi - q^2 - q & \text{for } i > q+1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

$$t_q(i) = \begin{cases} qi & \text{for } i \leq q+1 \\ 2q^2 - (i-2)q - 1 & \text{for } i > q+1 \end{cases},$$

$$t_{q+1}(i) = \begin{cases} q^2 + q - iq & \text{for } i \leq q+1 \\ 0 & \text{otherwise} \end{cases}.$$

Moreover for $i = 2q - 1$ we have

$$H(d) = -\frac{q^3 - q^2 + 2q - 2}{q^2 + q - 1}.$$

Remark 1.3. We do not know what happens for d such that $d \leq (q(d) - 1)^2 + q(d)$. The first such d is $d = 32$ with $q(32) = 7$. See the end of the last section.

This conjecture has been verified for $d \leq 10$ in [10]. In the present paper, we extend the range of the validity of the Conjecture to $d \leq 31$. This is our first main result. We repeat the results from [10] for completeness.

Theorem 1.4 (Values of absolute linear Harbourne constants). *For $2 \geq d \geq 31$ we have*

d	$H(d)$
2	0
3	-1
4	$-4/3 \approx -1.333$
5	$-3/2 = -1.5$
6	$-12/7 \approx -1.714$
7	-2
8	-2
9	$-9/4 = -2.25$
10	$-29/12 \approx -2.416$
11	$-33/13 \approx -2.538$
12	$-36/13 \approx -2.769$
13	-3

d	$H(d)$
14	$-54/19 \approx -2.842$
15	-3
16	$-16/5 = -3.2$
17	$-67/20 = -3.35$
18	$-24/7 \approx -3.428$
19	$-76/21 \approx -3.619$
20	$-80/21 \approx -3.809$
21	-4

d	$H(d)$
22	$-108/29 \approx -3.724$
23	$-115/30 \approx -3.833$
24	-4
25	$-125/30 \approx -4.166$
26	$-129/30 = 4.3$
27	$-135/31 \approx -4.354$
28	$-140/31 \approx -4.516$
29	$-145/31 \approx -4.677$
30	$-150/31 \approx -4.838$
31	-5

Table 1 Values of $H(d)$ for up to 31 lines

Remark 1.5. It comes as a surprise that the function $H(d)$ is not decreasing with d increasing.

We prove the Conjecture for any $d = q(d)^2 + q(d) + 1$, see Corollary 4.

The last assertion is a consequence of the following more general result.

Theorem 1.6 (Lower bound on Harbourne constants). *For $d \geq 6$ we have*

$$H(d) \geq -\frac{1}{2}\sqrt{4d-3} + \frac{1}{2}.$$

For $d = q^2 + q + 1$ with $q \in \mathbb{Q}$ we have the equality. In this case $H(d) = -q$ is computed by the configuration consisting of all lines in the finite projective plane $\mathbb{P}^2(\mathbb{F}_q)$.

Theorem 1.7 (Upper bound for Harbourne constants). *For $d \geq 7$ and with $r = r(d)$, we have*

$$H(d) \leq -2 \frac{r^4 + r^3 - r - (d-1)^2}{r^4 + 2r^3 - r - d^2 + d - 2}.$$

We will considerably improve this bound for some d in Proposition 5.1. In order to prove Theorem 1.4 we introduce some new tools, which might be of independent interest in other areas of combinatorics and geometry. We discuss also how our problem is related to the classical geometric problem on the existence of projective planes with certain numbers of points. Our investigations are accompanied by Singular [6] computations. The complete script of our program is provided in the Appendix.

2 Initial data

Let $\mathcal{L} = \{L_1, \dots, L_d\}$ be a configuration of lines in the projective plane $\mathbb{P}^2(\mathbb{K})$. Let t_k be the number of points where exactly k lines intersect. Then we have the following basic combinatorial equality

$$\binom{d}{2} = \sum_{k=2}^d t_k \binom{k}{2}. \quad (2)$$

Note that using this notation and taking into account (2) we can simplify the way $H(\mathbb{K}, \mathcal{L})$ is expressed:

$$H(\mathbb{K}, \mathcal{L}) = \frac{d^2 - \sum_{k=2}^d t_k k^2}{s} = \frac{d - \sum_{k=1}^s m_{\mathcal{L}}(P_k)}{s}. \quad (3)$$

Now our approach to computing or bounding Harbourne constants is based on the following idea. For a fixed d we consider the set \mathcal{T} of all integral solutions $T = (t_2, t_3, \dots, t_d)$ of the equality (2) and we compute the resulting combinatorial quotient

$$q(T) = \frac{d^2 - \sum_{k=2}^d t_k k^2}{\sum_{k=2}^d t_k}. \quad (4)$$

Of course not all elements of \mathcal{T} come from geometric configurations. So the task is to sort out those which cannot be obtained geometrically and then to find the minimum of $q(T)$'s for those which can.

3 Criteria for the nonexistence of a geometric configuration

In [10] we introduced a number of criteria to deal with this problem. Here we begin with a useful modification of what was called a two pencils criterion. We keep this name and hope that this will not lead to any confusion.

Lemma 3.1 (Two pencils criterion). *Let $\mathcal{L} = \{L_1, \dots, L_d\}$ be a configuration of lines in the projective plane $\mathbb{P}^2(\mathbb{K})$, with the singular set $\{P_1, \dots, P_s\}$, with $s \geq 2$. Let m_1, \dots, m_s be the multiplicities of points P_1, \dots, P_s respectively. Without loss of generality we can assume that*

$$m_1 \geq m_2 \geq \dots \geq m_s.$$

Then either

$$m_1 m_2 + 2 \leq s, \quad (5)$$

or, if (5) does not hold,

$$(m_1 - 1)(m_2 - 1) + a \leq s,$$

where a is equal to the minimal number of singular points lying on a line passing through P_1 and P_2 . The number a can be easily computed combinatorially.

Proof. Assume that points P_1 and P_2 do not lie on a configuration line, then the lines from two pencils (lines through P_1 , resp. P_2) meet in $m_1 m_2$ points. Together with P_1 and P_2 we get (5).

If (5) does not hold, points P_1 and P_2 lie on a configuration line. Lines from these two pencils (apart from the common line) meet in $(m_1 - 1)(m_2 - 1)$ points. Now we add the number of points on the common line, which is at least a . \square

The following Example illustrates how the two pencil criterion is applied

Example 3.2. The following data: $d = 10$, $t_3 = 7$ and $t_4 = 4$ is a solution of (2). Then $m_1 = m_2 = 4$ and $s = 11$. Since $4 \cdot 4 + 2 > 11$, we pass to the second inequality. Now $a = 3$, since the line through P_1 and P_2 meets with six other lines at these two points, hence there must be another point on this line (and one point of multiplicity 4 suffices). The inequality $3 \cdot 3 + 3 > 11$ shows that there is no geometrical configuration satisfying above data.

The next idea is to doubly count the incidences. First we need to introduce some notation. To a configuration line L we attach its *type vector*

$$\nu(L) = (\nu_2(L), \nu_3(L), \dots, \nu_d(L)),$$

where $\nu_k(L)$ denotes the number of points of multiplicity k on L . For example the line L in Figure 1 has type $\nu(L) = (1, 2, 0, 0, 0)$.

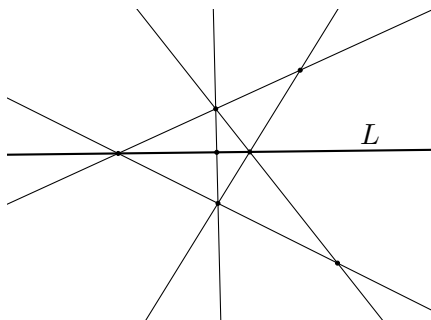


Figure 1

Now, let $n_\nu(\mathcal{L})$ be the number of lines in \mathcal{L} with the type vector ν . Then we have

$$\begin{cases} \sum_{\nu} n_\nu(\mathcal{L}) = d \\ \sum_{\nu=(\nu_2, \dots, \nu_d)} n_\nu(\mathcal{L}) \cdot \nu_k = k \cdot t_k, \quad \text{for } k = 2, \dots, d. \end{cases} \quad (6)$$

The first equation simply counts all lines in a configuration. The others count all “incidences” — a line passing through a point of given multiplicity k count as one incidence.

Let $T = \{t_2, \dots, t_d\}$ be a set of integers satisfying (2) for a fixed d . To these numbers there is the associated system of equations (6). In this system the symbols $n_\nu(\mathcal{L})$ are unknown. Which type vectors $\nu(L)$ can appear in the given configuration can be easily determined in advance. Their number is quite restricted. If the system (6) has no-negative solutions, then it follows that the set T cannot be realized geometrically.

However the set of equalities (6) is not always sufficient for our purposes. Let us assume that in a configuration there is a unique point of multiplicity m , that is $t_m = 1$. Then all lines passing through this point (that is all lines L with $\nu(L) = (\nu_2, \dots, \nu_{m-1}, 1, \nu_{m+1}, \dots, \nu_d)$) belong to the same pencil. Now for $k \neq m$ we count all points of multiplicity k on these lines, which obviously must be at most t_k ,

$$\sum_{\nu=(\nu_2, \dots, \nu_d), \nu_m=1} n_\nu(\mathcal{L}) \cdot \nu_k \leq t_k, \quad \text{for } k = 2, \dots, d, k \neq m. \quad (7)$$

So the new and powerful criterion for nonexistence works as follows: write down all equations (6) with the set of inequalities (7) for all $t_m = 1$, then try to solve this system of linear equations and inequalities in non-negative integers. This is a problem, well-known as integer programming, and there are many algorithms and software to deal with it.

Example 3.3. The following data: $d = 14$, $t_3 = 7$, $t_4 = 10$ and $t_5 = 1$ is a solution of (2). If it corresponds to a geometrical configuration then there are exactly four type vectors ν , for which $n_\nu(\mathcal{L})$ may be non-zero (a fixed line must meet with 13 other in singular points using only multiplicities appearing in the configuration, hence we can easily write down all possibilities). These are $(0, 5, 1, 0, \dots)$, $(0, 2, 3, 0, \dots)$, $(0, 3, 1, 1, \dots)$, $(0, 0, 3, 1, \dots)$. Assume that we have a (resp. b, c, d) lines in \mathcal{L} with resp. types. We have the following system of equalities:

$$\begin{cases} a + b + c + d = 14, \\ 5a + 2b + 3c = 21, \\ a + 3b + c + 3d = 40, \\ c + d = 5. \end{cases}$$

There are two nonnegative integer solutions, namely $(a, b, c, d) \in \{(0, 9, 1, 4), (1, 8, 0, 5)\}$. Observe however that $t_5 = 1$ allows us to use two additional inequalities

$$\begin{cases} 3c \leq 7, \\ c + 3d \leq 10. \end{cases}$$

The last inequality gives a contradiction, hence the initial data in this example does not come from any geometrical configuration.

3.1 A SINGULAR script

We wrote a SINGULAR script, which, given a number of lines d and a value $h := q(T)$ for a geometric configuration, works as follows:

- it enumerates first all possible arrays of integers $T = (t_2, \dots, t_d)$ satisfying (2),
- for each such an array it computes the quotient $q(T)$ as in (4),

- for those quotients, which satisfy $q(T) < h$ it checks whether the two pencils criterion works,
- if this is not the case, then the script produces an input for linear programming problem given by (6) and (7), then it uses a glpsol software to solve it,
- the results are reported; if for all T with $q(T) < h$ one of the two above criteria verifies the non-existence of a geometric configuration with data T , then h is a lower bound for $H(d)$. Otherwise the test fails and we do not know $H(d)$

The script is revoked by the command

```
check(number_of_lines, tested_bound, "output_file");
```

For example `check(10,-29/12,"result")` checks the validity of the number $H(10)$ provided in Theorem 1.4.

4 Proofs of the lower and upper bounds

In this section we prove Theorem 1.6 and Theorem 1.7. We begin with the lower bound.

Proof of Theorem 1.6. For $d \geq 6$ and $s \geq 1$ we consider the following function

$$f(d, s) = \frac{d}{s} - \frac{1}{2} - \frac{1}{2} \sqrt{1 + \frac{4d^2 - 4d}{s}}.$$

For a fixed field \mathbb{K} and positive integer d , let \mathcal{L} be a configuration of d lines with altogether s singular points. Then we have the following

Claim

$$H(\mathbb{K}, \mathcal{L}) \geq f(d, s). \quad (8)$$

Taking this for granted, Theorem 1.6 follows easily. Indeed, first of all the right hand side in (8) does not depend on \mathbb{K} , so that

$$H(d) \geq \min_{s \geq 1} f(d, s).$$

If \mathcal{L} is a pencil, i.e. $s = 1$, then $H(\mathbb{K}, \mathcal{L}) = f(d, 1) = 0$. Otherwise by the celebrated de Bruijn-Erdős Theorem [5] it must be $s \geq d$. Elementary calculus shows that for a fixed d the function $f(d, s)$ is strictly increasing for $s \geq d$. Hence finally

$$H(d) \geq \min_{s \geq d} f(d, s) = -\frac{1}{2} \sqrt{4d - 3} + \frac{1}{2}.$$

The extra assertion of Theorem 1.6 will be proved at the end of this section.

Now we turn back to the Claim.

Using (3) we have

$$H(\mathbb{K}, \mathcal{L}) = \frac{d}{s} - \frac{\sum_{k=1}^s m_{\mathcal{L}}(P_k)}{s},$$

so it suffices to show that

$$M := \frac{\sum_{k=1}^s m_{\mathcal{L}}(P_k)}{s} \leq \frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{4d^2 - 4d}{s}}. \quad (9)$$

The idea now is to apply Jensen's inequality (10) to (2).

Recall that for a convex function $\varphi(x)$ and non-negative numbers $\lambda_1, \dots, \lambda_s$ such that $\sum_{i=1}^s \lambda_i = 1$ there is

$$\sum_{i=1}^s \lambda_i \varphi(x_i) \geq \varphi\left(\sum_{i=1}^s \lambda_i x_i\right). \quad (10)$$

The function $\varphi(x) = x(x-1)$ satisfies the assumptions. Hence, from (2) we obtain with $\lambda_1 = \dots = \lambda_s = \frac{1}{s}$

$$\begin{aligned} \frac{1}{s}d(d-1) &= \frac{1}{s} \sum_{k=1}^s m_{\mathcal{L}}(P_k)(m_{\mathcal{L}}(P_k) - 1) \geq \\ &\geq \left(\frac{1}{s} \sum_{k=1}^s m_{\mathcal{L}}(P_k)\right) \left(\frac{1}{s} \sum_{k=1}^s m_{\mathcal{L}}(P_k) - 1\right) \\ &= M(M-1). \end{aligned} \quad (11)$$

It is elementary to check that this implies (9) and we are done. \square

Now we prove the upper bound.

Proof of Theorem 1.7. This bound is obtained in a rather naive way. Let $r = r(d)$. We consider the projective plane $\mathbb{P}^2(\mathbb{F}_r)$ as embedded in the projective plane defined over the algebraic closure $\overline{\mathbb{F}}_r$. Then \mathcal{L}_1 is the configuration of all $d_1 = r^2 + r + 1$ lines coming from $\mathbb{P}^2(\mathbb{F}_r)$. Then we take $d_2 = d - d_1$ *general* lines in $\mathbb{P}^2(\overline{\mathbb{F}}_r)$. These lines form another configuration \mathcal{L}_2 . Since they are general, they intersect pairwise in $\binom{d_2}{2}$ distinct points and they intersect the lines in \mathcal{L}_1 in $d_2 d_1$ distinct points. Thus for $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ we have

$$t_k(\mathcal{L}) = \begin{cases} \frac{1}{2}d_2(d_2 - 1) + d_1 d_2 & \text{for } k = 2 \\ r^2 + r + 1 & \text{for } k = r + 1 \\ 0 & \text{otherwise} \end{cases}. \quad (12)$$

The bound follows then computing the Harbourne constants and expressing everything in terms of d and $r = r(d)$. Note that for $d = r^2 + r + 1$, we get $H(d) \leq -r(d)$. \square

We conclude this section by showing the extra claim in Theorem 1.6.

Corollary 4.1. *Let $d = q^2 + q + 1$ with $q \in Q$. Then*

$$H(d) = -q$$

and $H(d)$ is computed by the configuration of all lines in $\mathbb{P}^2(\mathbb{F}_q)$.

Proof. The inequality $H(d) \leq -q$ follows from Theorem 1.7. The lower bound $H(d) \geq -q$ follows in turn from Theorem 1.6. Note additionally that the proof of Theorem 1.6 shows then that there is the equality in (9). Hence $d = s$ in this case and we conclude again by the de Bruijn-Erdős Theorem. Note that whereas the configuration consists of all all lines in $\mathbb{P}^2(\mathbb{F}_q)$, it might be embedded in some larger projective plane. \square

5 Results justifying Conjecture 1.2

We show first that there are infinitely many values of d such that

$$H(d) \leq h(d)$$

holds. More precisely we have the following result

Proposition 5.1. *Let d be a positive integer such that*

$$(q-1)^2 + (q-1) + 1 < d \leq q^2 + q + 1$$

with $q = q(d)$. Then

$$H(d) \leq h(d).$$

Proof. The idea is to construct a configuration of lines with invariants indicated in Conjecture 1.2. To this end let $i = q^2 + q + 1 - d$. Let \mathcal{L}_0 be the configuration of all lines in $\mathbb{P}^2(\mathbb{F}_q)$. For $i \leq q+1$, we fix a point $P_1 \in \mathbb{P}^2(\mathbb{F}_q)$ and remove exactly i lines passing through the point P_1 getting the configuration \mathcal{L}_1 . These lines intersect only at P_1 , so that with every line we decrease the number of $(q+1)$ -fold points by q and increase the number of q -fold points by q as well. The multiplicity m_1 of the point P_1 is $q+1-i$, whereas for $i = q$ and $i = q+1$, the point is no more a singular point of the configuration. It is then elementary to check that

$$H(\mathbb{F}_q, \mathcal{L}_1) = h(d). \quad (13)$$

For $q+1 < i \leq 2q-2$, we remove all $q+1$ lines passing through P_1 . This results in a configuration of q^2 lines with $q^2 + q$ points of multiplicity q . Then we remove the remaining $i - (q+1)$ lines from the pencil of lines passing through a second point P_2 . Counting as above, we get (13).

Finally for $i = 2q-1$, we fix two points P_1, P_2 and remove the line joining them, and $2(q-1)$ additional lines: $q-1$ from a pencil through P_1 and $q-1$ from the other pencil. This results in a configuration of $q^2 - q + 2$ lines with

$$\begin{aligned} t_{q+1} &= 1 \\ t_q &= 3(q-1) \\ t_{q-1} &= (q-1)^2 \end{aligned} \quad (14)$$

and all other $t_k = 0$. This gives $H(d) \leq h(d)$ also in this case. \square

Now we are in the position to prove Theorem 1.4.

Proof of Theorem 1.4. For all d between 2 and 31, Proposition 5.1 applies, so that $H(d)$ is at most equal to the numbers stated in Table 1. Turning to the lower bound it turns out that our SINGULAR script works in all cases. This ends the proof. \square

We pass now to d in the range $32 \leq d \leq 43$. For these values of d we have $q = q(d) = 7$, so consequently $d \leq (q-1)^2 + (q-1) + 1$. Hence the construction used in Proposition 5.1 does not apply. It is well known that there is no projective plane with 43 points (this would correspond to $q = 6$), see [4].

Of course $t_7 = 43$ and all other $t_k = 0$ is a solution to (2) with $d = 43$. Our program cannot exclude this configuration. It also cannot exclude any configuration resulting from this fake $\mathbb{P}^2(\mathbb{F}_6)$ configuration by removing lines. What it can is to

exclude any lower values of Harbourne constants. So that we can conclude that for $32 \leq d \leq 42$ it is

$$H(d) \geq h(d)$$

and

$$H(43) > -6.$$

It would be very interesting to modify our approach in a way opening access to configurations coming from fake projective planes. We hope to come back to this question in the next future.

Acknowledgement. This paper was to large extend written while the last author visited the University of Freiburg. It is a pleasure to thank Stefan Kebekus for hospitality. We would like also to thank Tomasz Szemberg for helpful conversations.

6 Appendix

```

ring R=0,x,dp;
proc writelistH(list l) {
  string s="H-const: "+string(l[1])+" conf: ";
  for (int k=2;k<=size(l);k++) {
    s=s+string(l[k])+" ";
  }
  return(s);
}
proc writelist(list l) {
  string s="";
  for (int k=2;k<=size(l);k++) {
    s=s+string(l[k])+" ";
  }
  return(s);
}
proc computeH(int n, list m) {
  number h=n*n;
  for (int i=2;i<=size(m);i++) {
    h=h-i*i*m[i];
  }
  int s=0;
  for (i=2;i<=size(m);i++) {
    s=s+m[i];
  }
  h=h/s;
  return(h);
}
proc twopencil(int n, list m) {
  int s=0;
  for (int i=2;i<=size(m);i++) {
    s=s+m[i];
  }
  string result="";

```

```

if (s<n) {return("  few points works");}
int r=2;
int p=size(m);
list tp;
while (r>0) {
  if (m[p]>0) {
    tp[r]=p;
    m[p]=m[p]-1;
    r=r-1;
  } else {
    p=p-1;
  }
}
if ((tp[1]-1)*(tp[2]-1)+2>s) {result=result+ "  TP works";} else
{
  int dp=2;
  int zp=n-(tp[1]-1)-(tp[2]-1)-1;
  while (zp>0) {
    if (m[p]>0) {
      zp=zp-(p-1);
      m[p]=m[p]-1;
      dp=dp+1;
    }
    else
    {
      p=p-1;
    }
  }
  if ((tp[1]-1)*(tp[2]-1)+dp>s) {result=result+"  TP(p) works";}
}
return(result);
}
proc writeconf(list l) {
  string s="  line conf: ";
  for (int k=2;k<=size(l);k++) {
    s=s+string(l[k])+" ";
  }
  return(s);
}
proc eqcritbyglp(int n, list m) {
  "checking "+writelist(m);
  "  conf for n="+string(n)+"...";
  list rm,nm,sm;
  for (int i=1;i<=size(m);i++) {rm[i]=0;}
  int p=0;
  int s;
  list cp;
  while (p<=size(m)) {
    rm[2]=rm[2]+1;
    p=2;
  }
}

```

```

while ((p<=size(m))&&(rm[p]>m[p])) {
    rm[p]=0;
    p=p+1;
    if (p<=size(m)) {rm[p]=rm[p]+1;} else {break;}
}
s=0;
for (i=2;i<=size(m);i++) {
    s=s+rm[i]*(i-1);
}
if (s==n-1) {
    cp[size(cp)+1]=rm;
}
}
int vrb=size(cp);
if (vrb==0) {return("    CONF(0) works");}
list eqs;
list eqq;
for (i=1;i<=size(cp);i++) {
    eqq[i]=1;
}
eqq[vrb+1]=n;
eqs[1]=eqq;
for (p=2;p<=size(m);p++) {
    if (m[p]>0) {
        for (i=1;i<=size(cp);i++) {
            rm=cp[i];
            eqq[i]=rm[p];
        }
        eqq[vrb+1]=p*m[p];
        eqs[size(eqs)+1]=eqq;
    }
}
int j;
string name=":w test";
write(name,"minimize value: a1");
name=":a test";
write(name,"subject to");
string wr;
for (i=1;i<=size(eqs);i++) {
    wr="e"+string(i)+" ":
    eqq=eqs[i];
    for (j=1;j<=vrb;j++) {
        if (j>1) {wr=wr+" ";}
        wr=wr+string(eqq[j])+" a"+string(j);
    }
    wr=wr+" = "+string(eqq[vrb+1]);
    write(name,wr);
}
int mm,k,o;
for (mm=2;mm<=size(m);mm++) {

```

```

    if (m[mm]==1) {
        for (i=2;i<=size(m);i++) {
            if ((i!=mm)&&(m[i]>0)) {
                wr="b"+string(mm)+"k"+string(i)+": ";
                o=0;
                for (k=1;k<=size(cp);k++) {
                    rm=cp[k];
                    if (rm[mm]==1) {
                        if (o==0) {o=1;} else {wr=wr+" + ";}
                        wr=wr+string(rm[i])+" a"+string(k);
                    }
                }
                wr=wr+" <= "+string(m[i]);
                if (o==1) {write(name,wr);}
            }
        }
    }
    write(name,"integer");
    for (j=1;j<=vrb;j++) {
        write(name," a"+string(j));
    }
    write(name,"end");
    int dummy=system("sh","glpsol --lp test -o solution");
    link solfile=":r solution";
    string sol=read(solfile);
    if (find(sol,"UNDEFINED",1)>10) {return(" SOLVER works");}
    return("");
}

proc throw(int n, list m) {
    string ii=writelstH(m);
    ii=ii+twopencil(n,m);
    if (find(ii,"works")==0) {ii=ii+eqcritbyglp(n,m);}
    return(ii);
}

proc scheck(int n, number bnd, string infofile, list m) {
    int ntp,ntpp,nsolver;
    int ok=1;
    infofile=":a "+infolfile;
    write(infolfile,"Input data: "+string(n)+" lines, bound for H-constant:
        "+string(bnd)+".");
    write(infolfile,"Configurations to exclude:");
    list v;
    string info;
    int na;
    for (int i=2;i<=n;i++) {
        v[i]=i*(i-1) div 2;
    }
    int sum=n*(n-1) div 2;
    list b;

```

```

for (i=2;i<=n;i++) {
    b[i]=sum div v[i];
}
int p;
int mm;
while (p<n) {
    m[3]=m[3]+1;
    p=3;
    while ((m[p]>b[p])||((p>n+1-mm)&&(p<mm))) {
        m[p]=0;
        p=p+1;
        m[p]=m[p]+1;
        if (p>mm) {mm=p;}
    }
    if (p==n) {break;}
    m[2]=sum;
    for (i=3;i<=n-1;i++) {
        m[2]=m[2]-v[i]*m[i];
    }
    if ((m[2]>=0)&&(m[n]==0)) {
        na=na+1;
        if ((na mod 10000)==0) {string(na)+" already checked...";writelst(m);}
        m[1]=computeH(n,m);
        if (m[1]<bnd) {
            info=throw(n,m);
            info;
            if (find(info,"works")==0) {ok=0;}
            write(infofile,info);
            if (find(info,"TP ")>0) {ntp++;}
            if (find(info,"TP(p)")>0) {ntpp++;}
            if (find(info,"SOLVER")>0) {nsolver++;}
        }
    } else {
        p=3;
        while (m[p]==0) {p=p+1;}
        m[p]=b[p];
    }
}
if (ok==1) {
    "All configurations have been excluded.";
    write(infofile,"All configurations have been excluded.");
}
info="TP used "+string(ntp)+" times, TP(p) used "+string(ntpp)+" times,
    SOLVER used "+string(nsolver)+" times.";
info;
write(infofile,info);
}
proc check(int n, number bnd, string infofile) {
    list m;
    for (int i=1;i<=n;i++) {

```

```

    m[i]=0;
}
scheck(n,bnd,infofile,m);
}
proc contcheck(int n, number bnd, string infofile, list sm) {
    list m;
    for (int i=1;i<=n;i++) {
        m[i]=0;
    }
    for (i=1;i<=size(sm);i++) {
        m[i]=sm[i];
    }
    scheck(n,bnd,infofile,m);
}

```

References

- [1] Barthel, G., Hirzebruch, F., Höfer, Th.: Geradenkonfigurationen und Algebraische Flächen. Aspects of Mathematics, D4. Friedr. Vieweg & Sohn, Braunschweig, 1987.
- [2] Bauer, Th., Di Rocco, S., Harbourne, B., Huizenga, J., Lundman, A., Pokora, P., Szemberg, T.: Bounded Negativity and Arrangements of Lines, International Mathematics Research Notices 2015 (2015), 9456–9471
- [3] Bauer, Th., Harbourne, B., Knutsen, A. L., Küronya, A., Müller-Stach, S., Roulleau, X., Szemberg, T.: Negative curves on algebraic surfaces, Duke Math. J. 162 (2013), 1877–1894
- [4] Bruck, R. H., Ryser, H., J.: The nonexistence of certain finite projective planes, Canadian J. Math. 1 (1949), 88–93
- [5] de Bruijn, N. G., Erdős, P.: On a combinatorial problem, Nederl. Akad. Wetensch., Proc. 51, (1948) 1277–1279 = Indagationes Math. 10, (1948) 421–423
- [6] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR 4-0-2 — A computer algebra system for polynomial computations. <http://www.singular.uni-kl.de> (2015).
- [7] Hirzebruch, F.: Arrangements of lines and algebraic surfaces. Arithmetic and geometry, Vol. II, 113–140, Progr. Math., 36, Birkhuser, Boston, Mass., 1983.
- [8] Mustață, M.: Multiplier ideals of hyperplane arrangements, Tran. Amer. Math. Soc. 358 (2006), no. 11, 5015–5023.
- [9] Pokora, P.: Harbourne constants and arrangements of lines on smooth hypersurfaces in $\mathbb{P}^3(\mathbb{C})$, to appear in Taiwan J. Math., DOI: 10.11650/tjm.19.2015.6338
- [10] Szpond, J.: On linear Harbourne constants, British Journal of Mathematics & Computer Science 8(4) (2015), 286–297.
- [11] Teitler, Z.: Multiplier ideals of general line arrangements in \mathbb{C}^3 . Comm. Algebra 35 (2007), 1902–1913.
- [12] <https://www.gnu.org/software/glpk/>

Marcin Dumnicki, Jagiellonian University, Institute of Mathematics, Łojasiewicza 6,
 PL-30-348 Kraków, Poland
E-mail address: Marcin.Dumnicki@im.uj.edu.pl

Daniel Harrer, Albert-Ludwigs-Universität Freiburg, Mathematisches Institut, D-79104 Freiburg, Germany

E-mail address: `daniel.harrer@math.uni-freiburg.de`

Justyna Szpond, Pedagogical University of Cracow, Institute of Mathematics,, Podchorążych 2, PL-30-084 Kraków, Poland

E-mail address: `szpond@up.krakow.pl`

Justyna Szpond current address: Albert-Ludwigs-Universität Freiburg, Mathematisches Institut, D-79104 Freiburg, Germany